

# Trabalho Prático de Programação Natural

## Cubo Mágico

Gabriel de Biasi<sup>1</sup>

<sup>1</sup> Departamento de Ciência da Computação  
Universidade Federal de Minas Gerais  
Av. Antônio Carlos, 6627 – Pampulha – Belo Horizonte – MG

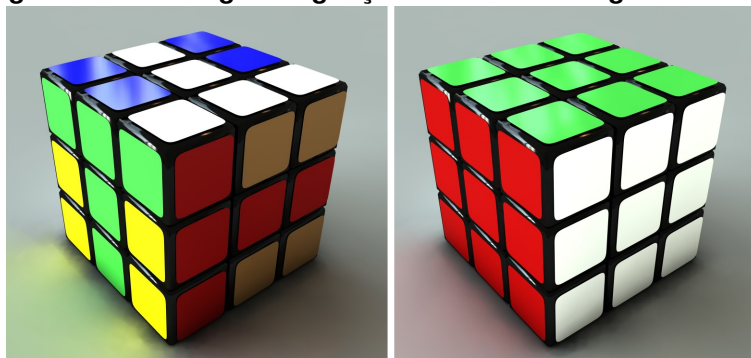
biasi@dcc.ufmg.br

### 1. Descrição do Problema

O brinquedo “Cubo Mágico” ou *Rubik’s Cube* foi criado por Rubik em 1974 e distribuído comercialmente em 1980. Possui um conjunto de 26 cubos menores, 6 faces com três tipos de cubos distintos: São 4 centros, 12 meios e 8 quinas. Cada tipo cubo possui um número específico de cores, onde os centros possui uma cor, meios possuem duas cores e as quinas possuem três cores.

É possível realizar movimentos em faces do cubo mágico sendo possível rotacionar os cubos desta face. Há 3 movimentos distintos que podem ser feitos em cada face, sendo eles: rotação horária, rotação anti-horária e dupla rotação. Logo, o cubo mágico tem um total de 18 movimentos possíveis. O objetivo do jogo é fazer com que todas as faces tenham a mesma cor. Na Figura 1 temos na esquerda um cubo em um estado “bagunçado” e outro no estado concluído.

Figura 1. Cubo mágico bagunçado e um cubo mágico resolvido



Neste trabalho, é proposto um algoritmo evolutivo que tenha capacidade de resolver uma dada instância de cubo mágico colocando-o no estado resolvido e ao mesmo tempo buscando minimizar a quantidade de movimentos necessários.

### 2. Metodologia

Para alcançar o objetivo do jogo, foi utilizado como base um método de resolução criado pelo professor *Morwen Thistlethwaite*, onde o espaço de buscas de soluções é categorizado e então o cubo precisa ser levado de uma categoria para a próxima utilizando apenas os movimentos permitidos da categoria atual [Scherphuis 2016].

## 2.1. Categorias do Cubo

Thistlethwaite criou 5 categorias, que descreve o tão próximo um cubo mágico está da solução. As categorias diminuem drasticamente o espaço de busca de soluções, fazendo com que a busca pela solução simplifique com o avanço entre as categorias.

**G0** Todos os estados do cubo mágico possíveis. Naturalmente, todos os cubos mágicos já estão presentes no conjunto G0. Sua ordem é de  $|G0| = 4.33 \times 10^{19}$ .

**G1** Nesta categoria, os meios do cubo estão devidamente **orientados**, ou seja, não são necessários os movimentos simples  $[L, R]$  para colocá-los em sua posição original. Sua ordem é de  $|G1| = 2.11 \times 10^{19}$ .

**G2** Na categoria G2 os meios da camada do meio não podem ser movidos de suas faces e a face de cima e a face de baixo só possuem suas respectivas cores amarelo/branco. Sua ordem é de  $|G2| = 1.95 \times 10^{10}$ .

**G3** Todas as faces opostas do cubo mágico agora possuem suas determinadas cores. As faces frente/atrás terão laranja ou vermelho, cima/baixo terão amarelo ou branco e esquerda/direita terão verde ou azul. Sua ordem é de  $|G3| = 6.63 \times 10^5$ .

**G4** Estado do cubo resolvido,  $|G4| = 1$ .

Os movimentos de cada categoria que permitem levar para a próxima sem a quebra de propriedade são os seguintes:

**Tabela 1. Movimentos permitidos em cada categoria**

<b>Categoria</b>	<b>Conjunto de Movimentos Permitidos</b>
G0	$(F, R, U, B, L, D)$
G1	$(F, R, U, B, L2, D2)$
G2	$(F, R, U2, B2, L2, D2)$
G3	$(F2, R2, U2, B2, L2, D2)$
G4	$\emptyset$

[El-Sourani et al. 2010].

## 3. Descrição Geral da Implementação

Nesta seção, será discutido como as funções básicas de algoritmos evolucionários foram implementadas para o contexto do problema proposto.

### 3.1. Fluxo de Trabalho do Algoritmo

O algoritmo inicia com a definição de três constantes, *Alpha* ( $\alpha$ ), *Theta* ( $\Theta$ ) e *Lambda* ( $\lambda$ ). Estas constantes representam, respectivamente, a quantidade de gerações máxima do algoritmo, o tamanho da população e a quantidade de indivíduos resolvidos necessária para avançar uma categoria.

### 3.2. Definição de um Indivíduo

Neste algoritmo, um indivíduo é representado por uma lista de movimentos no cubo mágico, partindo do estado do cubo que foi passado quando o algoritmo iniciou. Exemplo:

É importante ressaltar que cada indivíduo deste algoritmo evolucionário possui **quatro** valores de *fitness* diferentes, que é melhor explicado na Subseção 3.6.

**Tabela 2. Exemplos de Indivíduos**

ID	Conjunto de Movimentos
$I1$	$[F, R2, Ui, B, L2, \dots, D, Fi]$
$I2$	$[Di, Ui, R2, D2, U, Li]$
$I3$	$[U2, R2, Ui, D, F2]$
$I4$	$[\emptyset]$

### 3.3. Seleção

A seleção de indivíduos é feita por ranqueamento simples, onde todos os indivíduos da população atual são ordenados por *fitness* e os  $\lambda$  primeiros indivíduos são selecionados e considerados os candidatos para gerar a próxima geração.

Após a seleção, todos  $\lambda$  candidatos terão a probabilidade  $\frac{1}{\lambda}$  de serem escolhidos e então serem duplicados. Este processo se repete até que a nova população alcance a quantidade de  $\Theta$  indivíduos.

### 3.4. Mutação

Etc.

### 3.5. Função *clean*

Esta função é chamada toda vez que uma sequência de mutação é criada para um indivíduo. A fim de reduzir o número de movimentos necessários para resolver o cubo mágico, movimentos em sequência que não geram efeitos no cubo são removidos e movimentos que podem ser simplificados são alterados, sem perda de contexto final do cubo.

A seguir, temos três exemplos de simplificação que podem ser feitos em uma sequência de movimentos. Na Tabela 3 apenas o movimento  $F$  é apresentado, entretanto estas regras podem ser utilizadas para quaisquer movimentos do cubo mágico e em qualquer ordem.

**Tabela 3. Exemplos de Reduções de Movimentos**

Inicial	Motivo	Final
$[F, Fi]$	Não produz efeito	$[\emptyset]$
$[F, F]$	Torna-se um giro duplo	$[F2]$
$[F, F2]$	Torna-se um giro invertido	$[Fi]$

### 3.6. Função *fitness*

## 4. Execução dos Experimentos

lala

## 5. Conclusão

Neste trabalho.

## **Referências**

- [El-Sourani et al. 2010] El-Sourani, N., Hauke, S., and Borschbach, M. (2010). An evolutionary approach for solving the rubik's cube incorporating exact methods.
- [Scherphuis 2016] Scherphuis, J. (2016). Thistlethwaite's 52-move algorithm.